VISUAL BASIC

ศรันย์ ไมตรีเวช

วิธีเรียนลัด เกี่ยวกับดาต้าเบสในวิชวลเบสิก 4.0

นำเสนอวิธีการเรียนลัด เกี่ยวกับดาต้าเบสในวิชวลเบสิก 4.0

หัวใจสำคัญของเครื่องมือพัฒนาแอพพลิเคชันในปัจจุบันนั้น เห็นจะไม่พ้นเรื่องเกี่ยวกับดาต้าเบส และนับวันเครื่องมือพัฒนาฯทั้งหลายก็ยิ่งทวีความซับซ้อน เต็มไปด้วยรายละเอียดแยกย่อยเพิ่มขึ้นทุกที หากตะลุยอ่านกันตรง ๆ เหมือนสมัยเรียนประถม-มัธยม ก็คงไม่ทันกินล่ะครับ

ผมจะลองลำดับให้ว่าจุดแรกเราควรเริ่มที่ไหน และควรลงเอยที่งานของคุณเองได้อย่างไร หากจุดใดถูกทำให้กระชับ ไม่ครอบคลุมเนื้อความโดยรวม ผมก็จะกล่าวระบุลงไปว่าควรหาอ่านเพิ่มในหน้าช่วยเหลือโดยออนไลน์ที่ใดบ้าง นี่อาจเป็นเทคนิคหนึ่ง ไม่ใช่ทั้งหมด และยิ่งไม่ใช่ดีที่สุด ผมเพียงอยากลองเสนอสำหรับคนที่ชอบศึกษาค้นคว้าในแนวนี้ ความจริงสิ่งที่น่าสนับสนุนก็คือการพยายามค้นคว้าจากแหล่งอ้างอิงมากแห่งที่สุดเท่าที่จะเป็นไปได้ แต่ละแห่งจะมีจุดชัดและจุดมัวต่างกัน หากเราสามารถรวบรวมจุดชัดของทุกแหล่งไว้ในหัว ก็จะได้เปรียบคนอื่นที่ชัดบ้างมัวบ้าง

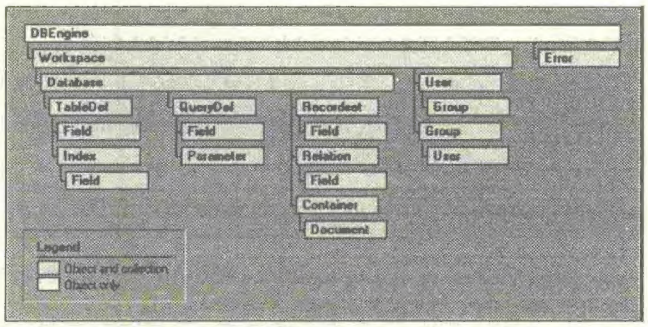
หนึ่งภาพดีกว่าหนึ่งพันคำจริงครับ เห็นตัวอย่างได้ชัดกันคราวนี้ หากคุณเริ่มคลำจากจุดอื่น เช่น อ่านตะลุยทีละลำดับ สิ่งที่คุณจะได้คือความสับสนงงงวย จับต้นชนปลายไม่ติด ได้หน้าลืมหลังลำดับเชื้อลำดับสายกันไม่ถูกต้องเริ่มกันด้วยภาพครับ อย่างที่เห็นในรูปที่ 1 คุณสามารถเปิดดูในหน้าช่วยเหลือของวิชวลเบสิก 4.0 เองได้ โดยใช้คีย์เวิร์ดคือ Database Engines และเลือก Topic คือ Microsoft Jet Database Engine 2.5/3.0

คุณสามารถคลิ๊กแต่ละรูปผืนผ้าของออบเจ็กต์ที่เห็นในผัง เพื่อเชื่อมโยงไปยังหน้าช่วยเหลือของออบเจ็กต์นั้น ๆ โดยตรง ด้วยวิธีนี้ คุณสามารถมองเห็นทั้งภาพรวม ความสายสัมพันธ์ระหว่างออบเจ็กต์ ลำดับชั้นจากบนไปหาล่าง รวมทั้งรายละเอียดต่าง ๆ ของออบเจ็กต์ทั้งหมดได้ครบถ้วน หากเริ่มที่อื่น เช่น หน้าช่วยเหลือของออบเจ็กต์ตัวใดตัวหนึ่งโดยเฉพาะ คุณจะพบว่ามันวกไปวนมาพิลึก เนื่องจากการทำความรู้จักกับออบเจ็กต์ตัวหนึ่งจะหลีกหนีการทำความรู้จักกับออบเจ็กต์ที่อยู่ข้างบนและที่อยู่ข้างล่างไม่ได้ ความสับสนจะเกิดขึ้นทั้งในระดับของการทำความรู้จักและระดับของการประยุกต์ใช้จริง

สำหรับคนที่เคยคุ้นกับดาต้าเบสออบเจ็กต์ในวิชวลเบสิก 3.0 รวมทั้งได้เคยมีประสบการณ์การทำงานกับไมโครซอฟต์แอ็กเซสมาก่อน รู้จักเจ็ตเอ็นจิน (Jet Engine) เป็นอย่างดี ก็จะรู้สึกว่าของใหม่ในวิชวลเบสิก 4.0 อย่างที่เห็นในรูปที่ 1 นั้นเป็นของดี น่าพิสมัย แต่หากไม่เคยมีประสบการณ์หรือความรับรู้เรื่องราวเกี่ยวกับเจ็ตเอ็นจินมาก่อนเลย อย่างนี้ก็แย่หน่อยครับ เพราะสิ่งที่ปรากฏตรงหน้านี้คือความซับซ้อนล้ำลึกเหลือจะให้อธิบายด้วยบทความเดียวสั้น ๆ โดยเฉพาะเมื่อคำนึงถึงการประยุกต์ใช้จริง

ออบเจ็กต์และคอลเลคชัน

คุณจะเห็นว่า DBEngine นั้นถูกแรเงาไว้ ในขณะที่ออบเจ็กต์อื่นที่อยู่ใต้ลงมาจะขาวโพลนไปหมด และมีนิยามกำกับไว้ที่มุมล่างด้านซ้ายมือว่าสี่เหลี่ยมขาวจะ



รูปที่ 1 ผังของกลุ่มออบเจ็กต์เกี่ยวกับดาต้าเบสในวิชวลเบสิก 4.0 ในหน้าช่วยเหลือออนไลน์ของวิชวลเบสิก 4.0

"เป็นได้ทั้งออบเจ็กต์และคอลเลคชัน" ส่วนสี่เหลี่ยมผืนผ้าที่ถูกแรเงาจะ "เป็นได้เพียงออบเจ็กต์อย่างเดียว"

ออบเจ็กต์ในวิชวลเบสิกทุกตัวถูกออกแบบ ถูกสร้างขึ้นมาให้มีหน้าที่อย่างใดอย่างหนึ่ง และเป็นออบเจ็กต์ที่มีความสัมพันธ์กับออบเจ็กต์อื่นในทางใดทางหนึ่ง จึงเป็นธรรมดาที่ต้องมีลำดับชั้น ว่าออบเจ็กต์ใดอาศัยออบเจ็กต์ไหนตั้งอยู่ อย่างเช่นที่ลูกนัยน์ตาคนต้องอาศัยศีรษะเป็นที่ตั้งเป็นต้น

คราวนี้มาดูคอลเลคชัน (Collection) อันนี้คือกลุ่มของออบเจ็กต์ชนิดเดียวกันครับ เหมือนอย่างนิ้วมือคนน่ะ เขาเรียกนิ้วเหมือนกัน แต่แบ่งได้เป็นนิ้วมือ นิ้วเท้า และก็สามารถเรียกแยกเป็นต่างหากจากกันได้อย่างละเอียดด้วย เช่นนิ้วชี้ของมือซ้าย หรือนิ้วนางของเท้าขวาอะไรอย่างนี้

คอลเลคชันจะประกอบขึ้นด้วยออบเจ็กต์ชนิดเดียวหลาย ๆ ตัว เหมือนอย่างนิ้วมือคน ในภาษาฝรั่งเขาจะเรียก fingers พอเห็นมี s เป็นตัวสุดท้ายห้อยหลัง เราก็จะรับรู้ทันทีว่าพูดถึงนิ้วหลาย ๆ นิ้ว อาจอยู่บนมือซ้ายหรือมือขวาก็ได้ มโนภาพเราจะไม่แยกจำเพาะลงไปว่าเป็นรูปหัวแม่มือ นิ้วชี้ นิ้วกลาง นิ้วนาง หรือนิ้วก้อย คอลเลคชันในวิชวลเบสิกก็เป็นอย่างนั้น เมื่อจะพูดถึงออบเจ็กต์ชนิดเดียวกันสักชุดหนึ่ง ยกตัวอย่างเช่น ฟิลด์ในเทเบิล ซึ่งคุณกับผมรู้กันว่ามีได้ตั้งแต่หนึ่งขึ้นไปจนนับร้อยฟิลด์ เขาก็จะมีคำเรียกให้กับหนึ่งหน่วยของฟิลด์ว่าเป็น Field และเรียกคอลเลคชันของมันว่าเป็น Fields

ความสำคัญ ณ จุดนี้มีอยู่ข้อเดียวที่คุณต้องจดจำให้ได้คือหากคำไหนมี s ตามหลัง อย่างนั้นเป็นคอลเลคชันแน่นอน ถ้าไม่มีก็แปลว่าเป็นออบเจ็กต์หนึ่งหน่วย สำหรับ DBEngine นั้น เป็นหนึ่งเดียวที่ไม่อาจมีฐานะคอลเลคชัน เนื่องจากเป็นตัวกลไกการติดต่อกับดาต้าเบสด้วยตัวเองเลยทีเดียว

ทั้งออบเจ็กต์และคอลเลคชันต่างก็มีพรอพเพอร์ตี้ส์ (Properties) หรือกล่าวอีกนัยหนึ่งคือข้อมูลซึ่งอยู่ภายในโมดูลของมันนั่นเอง และพรอพเพอร์ตี้ส์หลายตัวก็อาจเป็นออบเจ็กต์อีกชนิดหนึ่งได้ เหมือนเช่นที่ลูกตาเป็นวัตถุคนละชนิดกับศีรษะ แต่ก็อาศัยศีรษะเป็นที่อยู่ เอาล่ะครับทราบแค่นี้มาดูกันต่อเกี่ยวกับเรื่องของโครงสร้างออบเจ็กต์เกี่ยวกับข้อมูลในวิชวลเบสิก 4.0

ทำความเข้าใจเบื้องต้นกันแล้ว สมมติว่าเรากำลังอยู่ในหน้าช่วยเหลือตามรูปที่ 1 พร้อมกัน ผมจะพาคุณคลิ๊กไปตามลำดับเลย เริ่มแต่ชั้นแรกไล่ไปหาชั้นรอง

DBEngine

คำว่า Database Engine นั้น เราอาจคิดถึงมันได้ว่าเป็นกลไกในการเข้าถึงข้อมูล ซึ่งกลไกที่ว่านี้ดั้งเดิมเป็นของกลาง ยังไม่มีชื่อ แต่พอทำเป็นสินค้าออกมา ก็มีชื่อเฉพาะตามแบบการใช้งานต่าง ๆ เช่น Microsoft Access Database Engine และ Microsoft Visual BASIC Database Engine ซึ่งมีวิธีเข้าถึงข้อมูลเหมือนกัน รู้จักฟอร์แมตดาต้าเบสเหมือน ๆ กัน ทว่าผู้ใช้จะเห็นสภาพแวดล้อมแตกต่างกัน

DBEngine เป็นออบเจ็กต์ใหญ่ที่สุดในบรรดา DAO (Data Access Object) ด้วยกัน หากคุณรู้จักกับ Jet ก็ให้คิดว่ามันเป็น Jet เลยก็ได้ ไม่มีออบเจ็กต์เกี่ยวกับดาต้าเบสตัวใดใหญ่ไปกว่ามันอีก ทำนองเดียวกับที่เราไม่อาจคิดถึงออบเจ็กต์ตัวใดใหญ่ไปกว่าออบเจ็กต์ App ซึ่งเป็นตัวแทนของแอพพลิเคชันปัจจุบันทั้งตัว

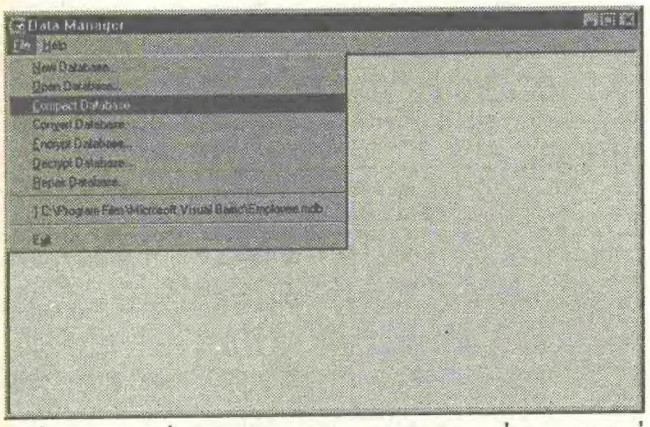
ความน่าสนใจของวิชวลเบสิก 4.0 จุดใหญ่ที่สุดเริ่มต้นที่ DBEngine เพราะเราสามารถควบคุม Jet ได้ทั่วถึงทั้งหมด ซึ่งหมายความว่าเราอาจทำได้ครอบคลุมทุกเนื้อหากิจกรรมเกี่ยวกับดาต้าเบส เหมือนอย่างเช่นสามารถทำได้ด้วยไมโครซอฟต์แอ็กเซส จะลดหลั่นลงมาบ้างก็น่าจะได้แก่เรื่องที่ไมโครซอฟท์จงใจแยกให้เป็นคุณสมบัติพิเศษเฉพาะของไมโครซอฟต์แอ็กเซสไม่กี่อย่าง

เพื่อมองให้เห็นออบเจ็กต์ DBEngine ได้กระจ่าง ลองย้อนไปดูแอพพลิเคชันแถมของวิชวลเบสิกซึ่งมีหน้าที่บริหารข้อมูลโดยเฉพาะ คือ Data Manager เชื่อว่าทุกคนเคยใช้แอพพลิเคชันตัวนี้กันมาแล้วทั้งนั้น เพราะมันอำนวยให้เราสร้างดาต้าเบส สร้างเทเบิล สร้างอินเด็กซ์ รวมทั้งแก้ไขปรับเปลี่ยนข้อมูลได้ เสมือนเป็น DBMS ย่อย ๆ ตัวหนึ่ง

ใน Data Manager หลายคนคงผ่านการใช้ฟังก์ชัน Compact Database มาแล้ว ฟังก์ชันนี้จะไม่ไปยุ่งเกี่ยวกับดาต้าเบสเดิมที่เราต้องการบีบให้เกิดความอัดแน่นกระชับ แต่จะคัดลอกไฟล์ดาต้าเบสเดิมไปสร้างเป็นไฟล์ดาต้าเบสใหม่ ซึ่งน่าหวังว่าเราสามารถเข้าถึงข้อมูลจำนวนมหาศาลได้ไวขึ้น ปรับเปลี่ยนแก้ไขข้อมูลได้คล่องตัวขึ้นกว่าของเดิม เนื่องจากพื้นที่ไร้ประโยชน์ในดาต้าเบสเก่าถูกกำจัดทิ้งไป (ดูรูปที่ 2)

การใช้ฟังก์ชันเช่น Compact Database ใน Data Manager นั่นแหละครับคือการสั่ง DBEngine โดยตรง ซึ่งนี่ก็หมายความว่าตั้งแต่เวอร์ชันเก่าของวิชวลเบสิก เราก็เคยผ่านการใช้ DBEngine กันมาแล้วโดยไม่จำเป็นต้องรับรู้เบื้องหน้าเบื้องหลัง หรือแม้แต่การปรากฏดำรงอยู่ของมัน แต่ในเวอร์ชันปัจจุบัน เรารู้แล้วครับ และสามารถบังคับบัญชามันได้โดยตรงในฐานของออบเจ็กต์ธรรมดาสามัญซึ่งมีพรอพเพอร์ตี้ส์และเมธอดให้เราดำเนินการใด ๆ

เช่นหากเราต้องการปรับปรุงดาต้าเบสให้อัดแน่นกระชับด้วยโค้ด ไม่ต้องพึ่งพา Data Manager ก็เพียงใช้ซินแท็กซ์นี้



รูปที่ 2 ฟังก์ชันเพื่อบีบดาต้าเบสให้อัดแน่น กระชับ เพื่อความเข้าถึงที่รวดเร็วขึ้นกว่าเดิม

DBEngine.CompactDatabase *ชื่อไฟล์ดาต้าเบสเก่า, ชื่อไฟล์ดาต้าเบสใหม่ [, นิพจน์กำหนดการเรียงลำดับอักขระ [, ค่าตัวเลือกพิเศษ]]*

DBEngine ยังให้โอกาสคุณเปลี่ยนเวอร์ชันจากเก่าเป็นใหม่ หรือจากใหม่ให้กลับเป็นเก่า ซึ่งอาจจำเป็นสำหรับคนที่ต้องติดตั้งแอพพลิเคชันตัวเดียวกันให้กับผู้ใช้หลายคน บางคนยังอยู่กับวินโดวส์ 3.x บางคนก็ขึ้นมาบนวินโดวส์ 95 แล้ว นอกจากนั้น คุณยังใช้มันตรวจเทียบลำดับและเข้ารหัสกลบเกลื่อนข้อมูล (Encryption) ได้โดยสั่งตรงมาจากโค้ด ไม่ต้องพึ่ง Data Manager เพียงคุณทำความรู้จักกับพรอพเพอร์ตี้ส์ของมันไม่กี่ตัวเท่านั้น

พรอพเพอร์ตี้ส์หลัก

1. DefaultUser ใช้กำหนดล่วงหน้าว่าผู้ใช้ที่จะล็อกเข้าเวอร์กสเปซปัจจุบันเป็นใคร

2. DefaultPassword ใช้กำหนดล่วงหน้าว่ารหัสผ่านผู้ใช้ที่จะล็อกเข้าเวอร์กสเปซปัจจุบันคืออะไร โดยมากมักจะมาควบคู่กันกับ DefaultUser และอาจกำหนดให้เป็นรหัสที่ "ผิด" คือถ้าเคาะเอ็นเตอร์เฉย ๆ จะเปิดดาต้าเบสไม่ได้

3. LoginTimeout ใช้กำหนดเวลาการล็อกอินดาต้าเบสบนเซอร์ฟเวอร์ผ่าน ODBC หากพ้นช่วงเวลาที่กำหนดไว้ในพรอพเพอร์ตี้ส์นี้จึงจะมีการฟ้อง Login Fail ให้ผู้ใช้รับทราบ พรอพเพอร์ตี้ส์นี้มีหน่วยเป็นวินาที โดยดีฟอลต์จะเป็น 20 วินาที

4. IniPath คุณสามารถควบคุม DBEngine ได้สะดวกมาก โดยอาจตั้งค่าเริ่มต้น เช่น ขนาดบัฟเฟอร์เป็นเท่าไหร่ ด้วยวิธีให้มีการอ่านไฟล์ .INI ชื่อใดชื่อหนึ่ง ซึ่งก็กำหนดได้ด้วยพรอพเพอร์ตี้ส์นี้เอง

5. Version บอกเวอร์ชันปัจจุบันของ DBEngine เป็นพรอพเพอร์ตี้ส์เอาต์พุต หมายความว่าคุณอ้างอิงได้อย่างเดียว จะไปปรับเปลี่ยนแก้ไขอะไรไม่ได้

เมธอดหลัก

1. CreateWorkspace สร้างเวอร์กสเปซขึ้นมารองรับการล็อกอินจากผู้ใช้ มีซินแท็กซ์คือ DBEngine.CreateWorkspace (ชื่อเวอร์กสเปซ, ชื่อผู้ใช้, รหัสผ่าน) ตัวอย่างเช่น

```
DBEngine.CreateWorkspace ("MyWorkspace", "Saran", "IamHere")
```

2. CompactDatabase ปรับปรุงคุณภาพดาต้าเบสให้ดีขึ้นด้วยการทำข้อมูลให้อัดแน่น ลดเนื้อที่สูญเปล่าลง เราสามารถเข้ารหัสกลบเกลื่อนข้อมูลและเปลี่ยนเวอร์ชันของ DBEngine ด้วยพรอพเพอร์ตี้ส์ตัวนี้ได้ด้วย

```
Private Sub Command1_Click()
    Dim WS As WorkSpace
    Dim DB As Database
    DBEngine.DefaultUser = "Admin"
    DBEngine.DefaultPassword = "WrongPassword"
    Set WS = DBEngine.CreateWorkSpace("WS")
    Set DB = WS.Opendatabase("MyDB.MDB")
End Sub
```

ตัวอย่างการสร้างเวอร์กสเปซและดาต้าเบส

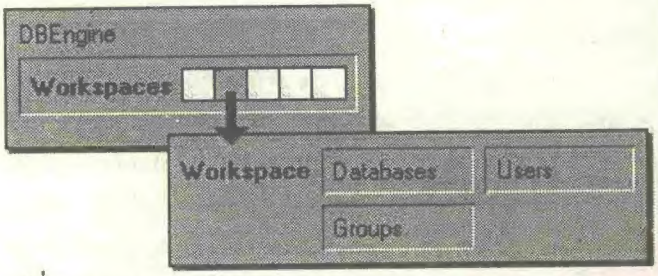
Workspace

เมื่อเราเปิดหรือสร้างดาต้าเบสขึ้นมาใหม่ ความจริงจะมีเวอร์กสเปซมารองรับ แต่เราอาจมองไม่เห็น และไม่รับรู้ถึงการปรากฏของมัน หากไม่ศึกษาใช้งานมัน ก็จะพลาดฟีเจอร์ที่เป็นประโยชน์ของ DBEngine ไป โดยเฉพาะอย่างยิ่งคุณที่ต้องเกี่ยวข้องกับผู้ใช้หลายคนในงานที่อาจมีการประสมประสานกันหลายดาต้าเบส

หากคุณคลิ๊กสี่เหลี่ยม Workspace ของแผนภาพในรูปที่ 1 ในหน้าช่วยเหลือของวิชวลเบสิก จะเห็นว่ามันนำไปยังอีกหน้าช่วยเหลือหนึ่งคือ Workspace และเห็นการแสดงขอบเขตความเป็น Workspace ดังรูปที่ 3

ดูจากรูป สิ่งที่เราเห็นในชั้นแรกคือขอบเขตของ DBEngine ภายในออบเจ็กต์ DBEngine จะประกอบไปด้วยคอลเลคชัน Workspaces (สังเกตว่ามี s) ซึ่งก็คือออบเจ็กต์ Workspace หลายตัวรวมกัน

ในชั้นที่สองจะซอยย่อยให้เห็นว่าแต่ละออบเจ็กต์เวอร์กสเปซประกอบไปด้วยคอลเลคชัน 3 ชนิด คือ Databases, Users และ Groups เพื่อไม่ให้หลงแนว ขอให้ลองย้อนกลับไปดูแผนภาพในรูปที่ 1 อีกที จะพบว่าทั้งสามคอลเลคชันที่กล่าวมานี้ อยู่ในลำดับชั้นเดียวกัน คือรองลงมาจากออบเจ็กต์เวอร์กสเปซทั้งสิ้น



รูปที่ 3 ขอบเขตความเป็น Workspace

จุดหลักที่จะกล่าวถึงได้แก่ดาต้าเบส ฉะนั้นเมื่อเห็นแล้วว่าที่แท้ดาต้าเบสก็เป็นเพียงออบเจ็กต์รองลงมาจากเวอร์กสเปซ ต้องอาศัยเวอร์กสเปซรองรับ ก็แปลว่าทุกครั้งที่มีการเปิดหรือสร้างออบเจ็กต์ดาต้าเบสขึ้นมา จะต้องมีการเลือกเวอร์กสเปซอันเป็นที่อยู่ด้วยเช่นกัน และไม่ว่าเราจะดำเนินการอย่างไรกับข้อมูลในดาต้าเบส ทุกเหตุการณ์จะยืนพื้นอยู่ในเวอร์กสเปซตัวใดตัวหนึ่งเท่านั้น หากเราเปิดดาต้าเบสขึ้นมาในเวอร์กสเปซที่ต่างกัน ก็แปลว่าจะสามารถทำงานได้อย่างเป็นอิสระจากกันอย่างเด็ดขาดไปด้วย เช่นเมื่อมีการใช้เมธอด BeginTrans, CommitTrans และ Rollback ก็ทำได้โดยไม่รบกวนกัน

ที่จุดนี้จำให้ดีว่าเมธอดเช่น BeginTrans, CommitTrans และ Rollback นั้นต่างก็เป็นเมธอดของ Workspace ไม่ใช่เมธอดของ DBEngine อย่างที่หลายคนเข้าใจ

ดังที่กล่าวแล้วในหัวข้อของ DBEngine ว่าเวอร์กสเปซเป็นสิ่งที่สร้างขึ้นได้ด้วยเมธอด CreateWorkspace ฉะนั้นก็คงเป็นที่เข้าใจแล้วว่าโดยดีฟอลต์ หากมีการเปิดหรือสร้างดาต้าเบสขึ้นมาใหม่ ก็จะมีการสร้างเวอร์กสเปซจาก DBEngine เช่นกัน แต่ถือเป็นอัตโนมัติ โดยเวอร์กสเปซที่ถูกสร้างโดยดีฟอลต์นี้จะมีหมายเลขอินเด็กซ์เป็น 0

พรอพเพอร์ตี้สำหลัก

1. Name กำหนดชื่อให้กับเวอร์กสเปซ เวลาอ้างถึงจะได้ใช้ชื่อแทนที่จะเป็นลำดับอินเด็กซ์ ขอให้สังเกตด้วยว่าปกติพรอพเพอร์ตี้ส์ Name จะกำหนดในโค้ดไม่ได้ แต่เมื่อเป็นตัวแปรออบเจ็กต์ ก็สามารถทำได้เนื่องจากเป็นสิ่งที่เพิ่งเกิดขึ้นหลังรัน ไม่ใช่ออบเจ็กต์ที่มีตัวตนตายตัวก่อนรันเหมือนอย่างเช่นออบเจ็กต์คอนโทรลทั่วไป

2. UserName เป็นชื่อผู้ใช้ซึ่งครอบครองเวอร์กสเปซปัจจุบัน อาจเป็นได้ทั้งอินพุตและเอาต์พุต

เมธอดหลัก

1. CreateDatabase เป็นฟังก์ชันใช้สร้างดาต้าเบส มีซินแทกซ์คือ

```
Set ชื่อตัวแปรดาต้าเบส = ชื่อเวอร์กสเปซ.CreateDatabase
(ชื่อดาต้าเบสพร้อมพาธ, ภาษา [, ค่าตัวเลือกพิเศษ])
```

ความจริงเราไม่จำเป็นต้องใส่ชื่อเวอร์กสเปซก็ได้ หากต้องการสร้างดาต้าเบสในดีฟอลต์เวอร์กสเปซ (หมายเลขอินเด็กซ์เป็น 0) สำหรับอาร์กิวเมนต์ที่ใช้กำหนดภาษานั้น ปกติจะใช้ค่าคงที่ dbLangeGeneral ซึ่งกำหนดให้เป็นภาษาอังกฤษ และรับภาษาไทยได้ในวินโดวส์ฉบับไทย ตัวอย่างเช่น

```
Set MyDB=CreateDatabase("C:\Temp\Try.mdb", dbLangGeneral)
```

แต่ถ้าหากต้องการระบุชื่อเวอร์กสเปซที่จะใช้สร้างดาต้าเบส ต้องไม่ลืมกำกับด้วยเลขอินเด็กซ์ด้วย เช่น

```
Set MyDB=Workspace(0).CreateDatabase("C:\Temp\Try.mdb",
dbLangGeneral)
```

ทั้งนี้ MyDB เป็นตัวแปรซึ่งถูกประกาศให้เป็นดาต้าเบสไว้ก่อน คือ Dim MyDB As Database ที่ส่วนประกาศระดับฟอร์ม (หากต้องการใช้ในทุกโพรซีเดอร์ของฟอร์ม) หรือบรรทัดใดบรรทัดหนึ่งภายในโพรซีเดอร์ปัจจุบันก่อนหน้าการใช้คำสั่ง Set นี้

2. OpenDatabase เป็นฟังก์ชันในการเปิดดาต้าเบส มีซินแทกซ์คือ

```
Set ชื่อตัวแปรดาต้าเบส = ชื่อเวอร์กสเปซ.OpenDatabase
(ชื่อดาต้าเบสพร้อมพาธ[, exclusive[, read-only[, source]]])
```

เหมือนกับเมธอด CreateDatabase หากต้องการเปิดดาต้าเบสขึ้นมาภายในดีฟอลต์เวอร์กสเปซ ก็ไม่จำเป็นต้องใส่ชื่อ เช่น

```
Set MyDB=OpenDatabase("C:\Temp\Try.mdb")
```

จะเห็นว่าเราไม่จำเป็นต้องใส่อาร์กิวเมนต์อื่นเข้าไป เนื่องจากเนื้อหาหลักในการเปิดดาต้าเบสอยู่ที่การระบุชื่อเท่านั้น เว้นแต่จะต้องการให้ดาต้าเบสเป็นอภิสิทธิ์กับผู้ใช้ซึ่งเข้าถึงด้วยโปรแกรมของเราเพียงคนเดียว หรือต้องการจำกัดสิทธิ์ไม่ให้ผู้ใช้ปรับเปลี่ยนข้อมูลก็กำหนดด้วยอาร์กิวเมนต์ exclusive และ read-only ตามลำดับ

3. BeginTrans ใช้เริ่ม Transaction ใหม่ การเปลี่ยนแปลงข้อมูลใด ๆ ที่เกิดขึ้นหลังบรรทัดนี้จะยังไม่ถูกจัดเก็บลงดาต้าเบสจริง

4. CommitTrans ใช้กำหนดจุดจบของ Transaction ปัจจุบัน และจัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นลงดาต้าเบสจริง บรรทัด CommitTrans ต้องมาทีหลัง BeginTrans เสมอ

5. Rollback ใช้กำหนดจุดจบของ Transaction ปัจจุบัน แต่แทนที่จะจัดเก็บการเปลี่ยนแปลงลงดาต้าเบส ก็จะย้อนกลับไปสู่ภาวะก่อนเกิด BeginTrans

เพื่อทำความเข้าใจกับการใช้เมธอดของ Workspace คุณต้องศึกษานิยามของ Transaction หนึ่ง Transaction นี่คือการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นตั้งแต่หนึ่งครั้งขึ้นไป คุณสามารถมองมันเป็น "กลุ่มการเปลี่ยนแปลง" ซึ่งการเปลี่ยนแปลงแต่ละครั้งหมายถึงสมาชิกแต่ละตัว หากเห็นว่าปฏิบัติการทั้งหมดที่เกิดขึ้นกับดาต้าเบสเป็นไปด้วยความเรียบร้อย ไม่มีอะไรผิดพลาดแน่แล้วค่อยทำให้ Transaction เกิดขึ้นอย่างเป็นทางการ (Commit Transaction ใช้เมธอด CommitTrans) แต่หากเห็นว่าปฏิบัติการอย่างใดอย่างหนึ่งล้มเหลว ก็ดึงการเปลี่ยนแปลงกลับที่เก่า (Roll Transaction Back ใช้เมธอด Rollback) ยกตัวอย่างเช่น เรามี Action Query อยู่สองตัว ซึ่งอาจทำได้สำเร็จหรือไม่สำเร็จ ก็เขียนเป็นโค้ดได้ดังนี้

```
Sub cmdAddOrder_Click ()

      On Error GoTo ErrorHandler
      DBEngine.WorkSpaces(0).BeginTrans '... (1)
            db.Execute "Insert Into Orders...  .",
dbFailOnError
            db.Execute "Insert Into [Order
Details]...." , dbFailOnError
      DBEngine.WorkSpaces(0).CommitTrans        '...(2)
      Msgbox "การเปลี่ยนแปลงเสร็จสิ้น"
      Exit Sub

ErrorHandler:
      Msgbox "การเปลี่ยนแปลงมีข้อผิดพลาด" & Err.Description
      DBEngine.Workspaces(0).Rollback   '...(3)
      Exit Sub
End Sub
```

ขอให้สนใจเฉพาะ 3 บรรทัดหมายเหตุที่ 1, 2 และ 3 ซึ่งมีการใช้กลุ่มเมธอดที่เกี่ยวกับ Transaction บรรทัดหมายเหตุที่ 1 ซึ่งใช้เมธอดBeginTrans จะบอกว่าถัดจากบรรทัดนี้จะเริ่ม Transaction ซึ่งคุณจะเห็นว่าเรากำหนดให้มีการเปลี่ยนแปลงกับเทเบิล Order และ Order Details ที่ตรงนี้จะเห็นว่าหนึ่ง Transaction นั้นมีได้หลายการเปลี่ยนแปลงอย่างไร

คราวนี้ถ้าโปรแกรมมัน "ลอง" ทำคำสั่ง Insert สำเร็จ ก็จะผ่านไปถึงบรรทัดหมายเหตุที่ 2 โดยไม่มีอะไรกระโตกกระตาก คือให้ CommitTrans ได้ และรายงานทางเมสเสจบ็อกซ์ด้วยว่าการเปลี่ยนแปลงเสร็จสิ้นเรียบร้อย

แต่หาก "ลอง" แล้วเกิดข้อผิดพลาด คุณก็ดักสั่งไว้แล้วตั้งแต่บรรทัดแรกของโพรซีเดอร์ปัจจุบัน ว่าให้ไปที่ ErrorHandler ซึ่งจะมีการสั่ง Rollback ในบรรทัดหมายเหตุที่ 3 ทุกอย่างก็จะเข้าสู่ภาวะปกติก่อน BeginTrans เหมือนไม่มีอะไรเกิดขึ้น

ก่อนจบ

ฉบับนี้ขอพักไว้ก่อน คราวหน้าจะมาต่อเรื่อง DAO และการประยุกต์ใช้จริงตัวต่อ ๆ ไป โดยเฉพาะออบเจ็กต์ Workspace นี่ยังว่าได้อีกยาวมากครับ สวัสดี

ในวันหนึ่งเมื่อนักพัฒนาซอฟต์แวร์ นักออกแบบฮาร์ดแวร์ และผู้จัดการแผนก จะต้องเดินทางไปประชุมต่างเมืองด้วยกัน ระยะทางที่ไปต้องผ่านภูเขาสูง ทางค่อนข้างลำบาก ในขณะที่ขับรถไปก็ได้เกิดเหตุการณ์ไม่คาด เบรครถเกิดเสียทำให้เบรคไม่อยู่ รถเสียหลักพุ่งชนขอบถนน เสียงร้องเอะอะโวยวายฟังไม่ได้ศัพท์ แต่โชคดีที่พวกเขาไม่ได้รับบาดเจ็บ รถบุบและเสียศูนย์ทำให้ล้อส่ายไปมา ซึ่งตอนนั้นปัญหามีอยู่ว่าพวกเขากำลังอยู่แค่ครึ่งทาง คุณคิดว่าพวกเขาจะทำยังไง ?

ผู้จัดการแผนก : "ผมรู้แล้ว ! และตอนนี้ผมคิดว่าเราควรจะประชุมกันเพื่อเสนอความคิดเห็น หาหลักการสรุปแนวทางเลือกทางออกที่ดีที่สุด พร้อมทั้งนำไปเป็นทางแก้ไขต่อไป หลังจากที่เราสามารถแก้ไขปัญหาเสร็จแล้ว เราก็จะสามารถเดินทางต่อไปได้"

นักออกแบบฮาร์ดแวร์ : "ไม่ต้องหรอกครับ ผมว่ามันนานเกินไป พอดีผมมีมีดพับวิคทอรีน็อคซ์ (มีดพับแมค-ไกเวอร์) ติดตัวมาด้วย แบบแผนเกิดขึ้นไม่ได้หากไม่ลองปฏิบัติก่อน เดี๋ยวผมจะลองซ่อมระบบเบรคดู แยกส่วนที่เสียหายและซ่อมมัน ผมคิดว่าเราก็ไปต่อได้แล้ว"

นักพัฒนาซอฟต์แวร์ : "เดี๋ยว ๆ ก่อนครับ ก่อนที่จะทำอะไร ผมคิดว่าเราน่าจะดันรถขึ้นไปบนถนน และดูว่าเหตุการณ์เมื่อกี้นี้มันเกิดขึ้นมาได้อย่างไร ขอวน Loop ดูอีกครั้งเถอะครับ"